

Aufgabe SBB Datenaustausch

Teil 1

Für eine Applikation die mit der SBB Daten austauscht muss ein entsprechendes **XML-Schema** definiert und ein **Beispiel XML** erstellt werden. Folgende Daten für eine **Verbindung** liegen vor:

Verbindung	Beispieldaten
Datum	2014-12-01
Von	
Bahnhof	St.Gallen
Zeit	12:11:00
Gleis	1
Nach	
Bahnhof	Zürich
Zeit	13:23:00
Gleis	16
Reise mit	ICN
Belegung	
Erste Klasse	1
Zweite Klasse	1
Bemerkung	Richtung: Genève

Die folgenden **Restriktionen** sind zu zusätzlich beachten:

- Das Feld "Bahnhof" muss minimal die Länge 2 und maximal die Länge 150.
- Das Feld "Gleis" kann nur eine Zahl grösser gleich 1 sein.
- Das Feld "Reise mit" muss minimal die Länge 2 und maximal die Länge 20.
- Die Felder der Belegung (1. und 2. Klasse) können nur die Werte 1,2 oder 3 verwendet werden.
- Das Feld "Bemerkung" ist das einzige Feld welches optional ist, d.h. es muss nicht zwingend vorhanden sein. Wenn es vorhanden ist darf das Feld maximal die Länge von 200 Zeichen haben.

Teil 2

Es sollen zwei XML-Parser geschrieben werden die XML-Files die dem Schema entsprechen einlesen können.

Parser 1

Technologie: **SAX** (Version 2.0)

Parser 2

Technologien: **DOM**

Beide Parser sollen in **Java** geschrieben werden und mittels **JUnit** sollen entsprechende Test-Klassen entstehen welche das korrekte Verhalten mittels unterschiedlichen **Szenarien überprüft**.

In beiden Fälle soll das/die selbe Daten-Objekt(e) entstehen in diesem die Werte der Verbindungsdaten abgelegt sind und welches für die Weiterverarbeitung (innerhalb des Java-Codes) genutzt werden kann.

Eingesetzte Technologien

- Java JDK 1.7 oder 1.8
- JUnit 4.x
- Eclipse Entwicklungsumgebung (Versionen: Indigo, Juno, Kepler, Luna)
- Code-Konvention
 - Sun/Oracle Code Conventions
 - <http://www.oracle.com/technetwork/java/javase/documentation/codeconvtoc-136057.html>
 - Ausnahme Zeilenlänge max. 200 Zeichen

Abgabe

Abgegeben soll ein ZIP-File in dem alle notwendigen XML-Dateien, XML-Schema, Klassen und Libraries (z.B. junit.jar) enthalten sind. Es kann das Eclipse-Projekt in das ZIP gepackt werden.

Informationen

Grundlagen und Begriffe werden im Kapitel 2 des Buchs "Java und XML" beschrieben.

Das Document Object Model (DOM)

DOM ist eine Entwicklung des W3C und wird von vielen Programmiersprachen unterstützt. Das Standard-DOM ist so konzipiert, dass es unabhängig von einer Programmiersprache ist und eine strikte Hierarchie erzeugt. DOM definiert eine Reihe von Schnittstellen, die durch konkrete Programmiersprachen implementiert werden.

- Offizielle Dokumentation von Oracle zum Thema DOM für Java:
<http://docs.oracle.com/javase/tutorial/jaxp/dom/index.html>
- Wikipedia zu DOM: http://de.wikipedia.org/wiki/Document_Object_Model
- Im XML-Buch "Java und XML" Kapitel 3

Simple API for XML Parsing (SAX)

SAX ist zum schnellen Verarbeiten der Daten von David Megginson als Public Domain entworfen worden. SAX ist im Gegensatz zu DOM nicht so speicherhungrig, weil das XML-Dokument nicht vollständig im Speicher abgelegt ist, und daher auch für sehr große Dokumente geeignet. Da SAX auf einem Ereignismodell basiert, wird die XML-Datei wie ein Datenstrom gelesen, und für erkannte Elemente wird ein Ereignis ausgelöst. Dies ist aber mit dem Nachteil verbunden, dass wahlfreier Zugriff auf ein einzelnes Element nicht ohne Zwischenspeicherung möglich ist.

- Offizielle Dokumentation von Oracle zum Thema SAX für Java:
<http://docs.oracle.com/javase/tutorial/jaxp/sax/index.html>
- Wikipedia zu SAX: http://de.wikipedia.org/wiki/Simple_API_for_XML
- Im XML-Buch "Java und XML" Kapitel 4